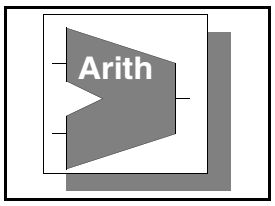
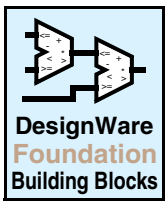




DW02_mult_6_stage

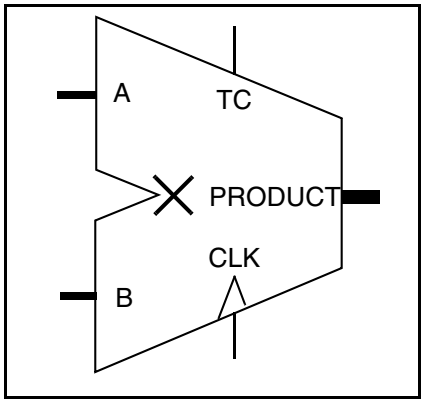
Six-Stage Pipelined Multiplier

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History

- Parameterized word length
- Unsigned and signed (two's-complement) data operation
- Six-stage pipelined architecture
- Automatic pipeline retiming



Description

DW02_mult_6_stage is a six-stage pipelined multiplier. DW02_mult_6_stage multiplies the operand A by B to produce a product (PRODUCT) with a latency of five clock (CLK) cycles.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*A_width* bits	Input	Multiplier
B	*B_width* bits	Input	Multiplicand
TC	1 bit	Input	Two's complement ■ 0 = Unsigned ■ 1 = Signed
CLK	1 bit	Input	Clock
PRODUCT	*A_width* + *B_width* bits	Output	Product (A × B)

Table 1-2 Parameter Description

Parameter	Values	Description
A_width	≥ 1	Word length of A
B_width	≥ 1	Word length of B

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Area or delay optimized flexible architecture	DesignWare

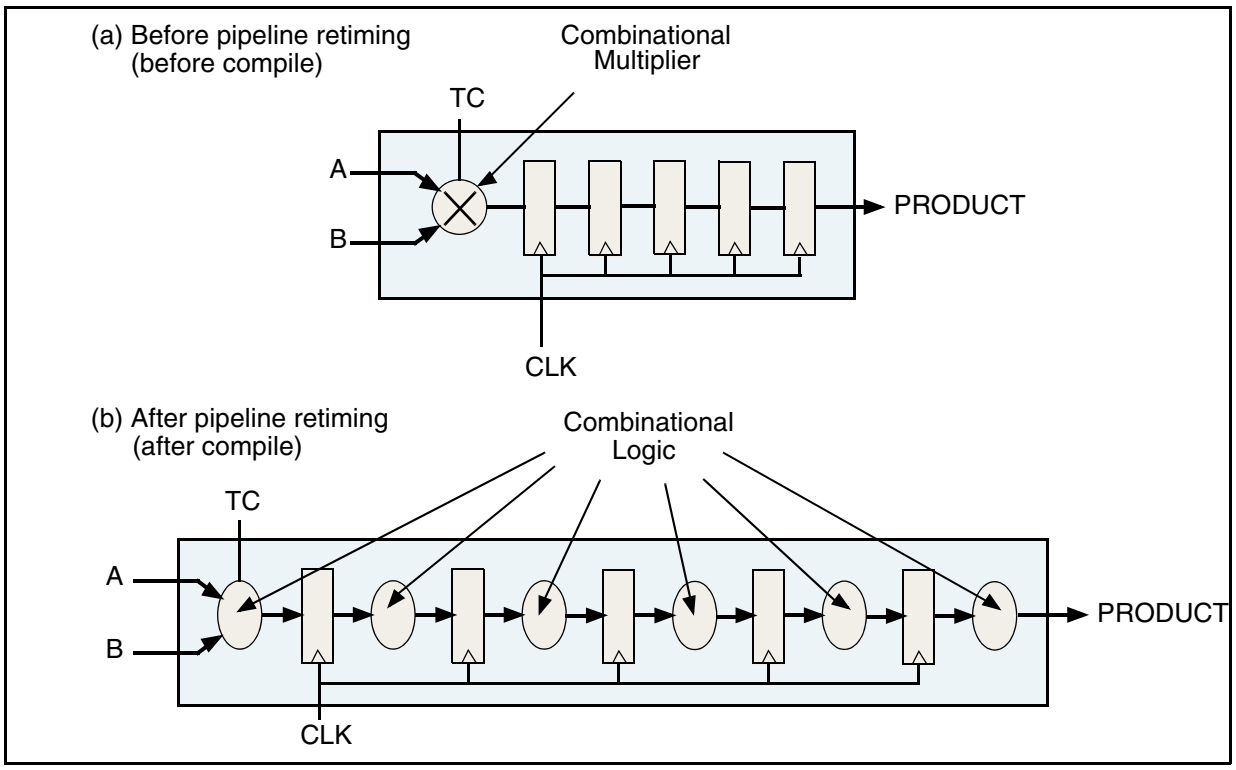
Table 1-4 Simulation Models

Model	Function
DW02.DW02_MULT_6_STAGE_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW02_mult_6_stage_sim.vhd	VHDL simulation model source code
dw/dw02/src_ver/DW02_mult_6_stage.v	Verilog simulation model source code

The control signal TC determines whether the input and output data is interpreted as unsigned (TC = 0) or signed (TC = 0) numbers.

Automatic pipeline retiming ensures optimal placement of pipeline registers within the multiplier to achieve maximum throughput.

Figure 1-1 Block Diagram



Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW02_mult_6_stage_inst is
  generic ( inst_A_width : POSITIVE := 8;
            inst_B_width : POSITIVE := 8 );
  port ( inst_A : in std_logic_vector(inst_A_width-1 downto 0);
         inst_B : in std_logic_vector(inst_B_width-1 downto 0);
         inst_TC : in std_logic;
         inst_CLK : in std_logic;
    PRODUCT_inst : out std_logic_vector(inst_A_width+inst_B_width-1 downto 0)
       );
end DW02_mult_6_stage_inst;

architecture inst of DW02_mult_6_stage_inst is
begin

  -- Instance of DW02_mult_6_stage
  U1 : DW02_mult_6_stage
    generic map ( A_width => inst_A_width, B_width => inst_B_width )
    port map ( A => inst_A,   B => inst_B,   TC => inst_TC,
               CLK => inst_CLK,   PRODUCT => PRODUCT_inst );
end inst;

-- pragma translate_off
configuration DW02_mult_6_stage_inst_cfg_inst of DW02_mult_6_stage_inst is
  for inst
  end for; -- inst
end DW02_mult_6_stage_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW02_mult_6_stage_inst( inst_A, inst_B, inst_TC,
                               inst_CLK, PRODUCT_inst );

  parameter A_width = 8;
  parameter B_width = 8;

  input [A_width-1 : 0] inst_A;
  input [B_width-1 : 0] inst_B;
  input inst_TC;
  input inst_CLK;
  output [A_width+B_width-1 : 0] PRODUCT_inst;

  // Instance of DW02_mult_6_stage
  DW02_mult_6_stage #(A_width, B_width)
    U1 ( .A(inst_A),   .B(inst_B),   .TC(inst_TC),
         .CLK(inst_CLK),   .PRODUCT(PRODUCT_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Removed the “Disabling Clock Monitor Messages” section
October 2019	DWBB_201903.5	■ Updated description of ‘str’ implementation in Table 1-3 on page 2 ■ Added the “Disabling Clock Monitor Messages” section
January 2019	DWBB_201806.5	■ Updated example in “HDL Usage Through Component Instantiation - VHDL” on page 3 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

© 2022 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com